

Remarks/Arguments

This Amendment and the following remarks are intended to fully respond to the final Office Action mailed December 15, 2008. With this Amendment, claims 1 and 5-18 are canceled without prejudice or disclaimer. Claims 39-50 are added and remain pending. Reconsideration and allowance are requested for the following reasons.

Claim Rejections – 35 U.S.C. § 103

Claims 1 and 5-18 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,560,774 to Gordon et al. (hereinafter “Gordon”) in view of “ASP.NET Configuration Files” to Sully (hereinafter “Sully”) and “Special Edition Using Microsoft ASP.NET” to Leinecker (hereinafter “Leinecker”).

Applicants respectfully traverse this rejection. However, in an effort to advance this Application to allowance claims 1 and 5-18 are canceled in favor of new claims 39-50. Reconsideration is requested for the following reasons.

Claim 39 is directed to one or more processor-accessible storage media comprising processor-executable instructions that, when executed, direct a device to perform file compilation actions. Claim 39 recites, in part, instantiating a single instance of a build provider host, wherein the build provider host includes a plurality of interfaces, wherein each of the plurality of interfaces is associated with one of the plurality of build providers during file compilation actions. Claim 39 further recites sequentially calling a generate code interface of each build provider, wherein each build provider contributes at least a portion of the source code of the build provider’s associated file to be compiled via one or more of:

- writing the source code to a code file object by calling a create code file object interface of the build provider host;
- writing the source code to a stipulated path by calling a get code file path interface of the build provider host; and
- generating a code compile unit by calling a get code object model provider interface and an add code compile unit interface of the build provider host;

In examples provided in the specification of the present application that are consistent with that of claim 39, various features and processes are disclosed to provide a pluggable build

architecture that enables extensibility for new file types via build provider interfaces of a build provider host and/or a build provider manager. See Application pg. 6, ll. 23 - pg. 7, ll. 3; Fig. 3; Fig. 4. For example, in one non-limiting embodiment, a build provider host 204 includes a create code file object interface 406 that creates a file object that is to include new source code for compilation 110. Id. at pg. 10, ll. 8-9. Additionally, the build provider host 204 includes a get code file path interface 408 that returns a path to a file whose source code is to be included in compilation 110. Id. at pg. 10, ll. 16-17. Further, the build provider host 204 includes a get code object model provider interface 410 that returns a code object model provider that can be used by build provider 208 to generate a code compile unit. Id. at pg. 11, ll. 1-2. The respective interfaces facilitate actions and/or communications between the build provider host 204 and multiple build providers 208 during compilation. Id. at pg. 9, ll. 23 - pg. 10, ll. 1.

Gordon, Sully, and Leinecker, either alone or in combination, fail to disclose or suggest the noted limitations of claim 39.

Rather, as noted in the Action, Gordon discloses a system including an execution engine 200 having a verifier 202, in which multiple source code language sources are compiled by compilers 204 into IL code. The verifier 202 verifies the IL code as well as metadata of the IL code, and the execution engine 200 compiles, interprets, *inter alia*, the IL code into executable code. See col. 6, ll. 8-57 and col. 27, ll. 11- col. 28, ll. 7. Gordon fails to disclose instantiating a build provider host, or sequentially calling a generate code interface of each build provided, as required by claim 39.

Sully also fails to disclose the noted limitations of claim 39. As noted in the Action, Sully simply discloses a configuration file used in the .Net Framework to define compilation settings for use when the .NET framework dynamically compiles resources, where typically there are several compilers setup for VB, C# and JavaScript source language. Sully also fails to suggest instantiating a build provider host, or sequentially calling a generate code interface of each build provided, as required by claim 39.

Leinecker simply describes techniques of teaching developers how to quickly develop applications in the context of .NET framework and distributed application development. See Chapter 1, Introduction. Leinecker does not disclose or suggest the noted limitations of claim 39.

The combination of Gordon, Sully and Leinecker, either alone or in combination, fails to teach or suggest all of the limitations of claim 39. Reconsideration and withdrawal of the rejection of claim 39, as well as claims 40-50, that depend therefrom, are therefore requested.

Conclusion

Favorable reconsideration is requested. Please contact the undersigned with any questions regarding this application.

Respectfully submitted,
MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, Minnesota 55402-0903
(612) 332-5300

Date: May 15, 2009

/Robert A. Kalinsky/
Name: Robert A. Kalinsky
Reg. No.: 50,471